REMARKS

Claims 1-21 were pending. New claims 22-29 are added.

Claims 1, 14, 18 and 22 are independent. Claims 14 and 22 are, respectively, method versions of apparatus claims 1 and 18.

Claims 2-13 depend on claim 1. Claims 15-17 and 26-29 depend on claim 14. Claims 19-21 depend on claim 18. Claims 23-25 depend on claim 22.

The following claims are amended: 1-21.

Table of Contents:

1. Claim Rejections – 35 USC § 102(b)

The Examiner rejects claims 1-6, 9-11, and 13-21 under 35 USC § 102(b) as being anticipated by European Patent 0 651 343 A1 (Butts).

Regarding claim 1, applicants direct the Examiner's attention to it being limited to an emulation board with a processor that executes "programmed instructions for configuring the first reconfigurable resource."

Butts is clearly directed to an emulation system where only the host computer is capable of executing programmed instructions. The Examiner has pointed to Figure 1 of Butts, with its "host interface" and "config. system," as presenting examples of a programmed processor (see Office Action mailed Apr. 12, 2007, rejection of claim 4 on page 3). However, applicants refer the Examiner to the following portions of Butts that clearly show this is not the case.

Section 1.4 of Butts ("Configuration") states the following:

> **[0241]** As described in the section on logic and interconnect chip technology, the configuration bit patterns for each chip are generated by the ERCGA netlist conversion tool. The final stage of the Realizer design conversion system collects the data from the configuration files generated for all chips into a single binary configuration file for the design, which is permanently stored in the host computer.
>
> **[0242]** Before each use of the Realizer system, its

> logic and interconnect chips are configured for the design to be used, by reading data from the configuration file, transferring it into the Realizer hardware through the host interface, and loading it into the chips. Configuration connections are provided between the host interface and all logic and interconnect chips in the system.

Butts, pages 30-31. In ¶ 0241 it is made clear that a single binary configuration file is generated on the host computer. In ¶ 0242 it is made clear that such configuration file is simply transferred through the host interface and into the reconfigurable chips. The last sentence, of the above quote from ¶ 0242, refers to the configuration system as merely "connections [that] are provided between the host interface and all logic and interconnect chips in the system."

To better appreciate the following quotes from Butts, it is useful to understand the types of suitable reconfigurable chips that Butts discusses. For this information, the Examiner is referred to the following sections of Butts: 1.1.1 ("Logic Chip Devices") and 1.1.2 ("Interconnect Chip Devices"). The type of chip focused upon, for both logic and interconnect, is referred to as a "Logic Cell Array" (LCA).

Further review of section 1.4 of Butts confirms that the "configuration system" is little more than connections between the host interface and the reconfigurable chips. For example, page 31 of Butts states the following:

> **[0244]** A unique configuration connection between each LCA and the host interface is not provided, as a system can have up to 3520 total logic and crossbar chips. Instead, there is a configuration bus, consisting of a multi-bit data path and a configuration clock, which is connected to all boards which have LCAs. Logic and crossbar chips are grouped for the purposes of configuration, with as many chips per group as there are bits in the data path. All chips in one group are configured in parallel.
>
> **[0245]** As shown in Fig. 40, each LCA in a group has its configuration data input connected to a different bit of the bus data path.

The above quotes, of ¶ 0244 and ¶ 0245, along with Figure 40, make it clear that the configuration system is just a "configuration bus" (see Figure 40) with some "config. control logic" (see Figure 40).

Section 1.5.1 of Butts ("Host Interface Architecture") states the following about the host interface:

> **[0248]** The Realizer system host interface is built along entirely conventional lines (Fig. 41). It consists of the host interface bus controller, the configuration bus controller, the clock generator and the reset controller.

Butts, page 31. A bus controller (i.e., the "host interface bus controller" or "configuration bus controller" referred to in the above quote), when constructed along "conventional lines," is always implemented with simple hardware and never with a programmed processor. Furthermore, ¶ 0255 of Butts, page 32, states that the "bus interface control ... consists of a finite state machine ... in an entirely conventional manner."

Like claim 1, all other independent claims presented by the applicant (i.e., claims 14, 18 and 22), are also limited to a "processor" that executes "programmed instructions" for "configuring the first reconfigurable resource" (which we shall also refer to as a "programmed processor"). Since Butts provides no teaching, suggestion or motivation for the use of a programmed processor, claims 14, 18 and 22 are also allowable over Butts for at least the same reasons as presented above for claim 1.

Since claims 1, 14, 18 and 22 are allowable, and all other pending claims are dependent upon these claims, all other pending claims are allowable for at least the same reasons.

In addition, regarding claim 9, Butts provides no teaching, suggestion or motivation for having a programmed processor on the same chip as the reconfigurable resource.

Because the prior art does not teach, suggest or motivate the use of an on-board programmed processor, one of ordinary skill in the art could not even consider the advanced functionality that such programmed processor makes possible. For example, claim 10 is directed to performing, on the emulation board itself, confirmation of a configuration of a reconfigurable resource. Claim 12 is directed to the use of software modules, with the programmed processor, such that a second software module controls invocation of a first software module, while the first software module itself has the programmed instructions for doing the configuration. Also, as is claimed in claim 19, a programmed processor on each emulation board permits first and second emulation boards to distributively configure their reconfigurable resources. Claim 20 adds the further limitation, to claim 19, that each emulation board have at least the following two software modules: one module that performs invocation and another module that does the configuration. Claim 21 adds the limitation that provision of programmed instructions, by the computer to the memory of the emulation board, is staged.

Claim 9 addressed the subject matter of "on-chip" processing when the present application was filed. Examples of where support can be found in the specification, for claims 10, 12, 19, 20 and 21, is provided by the following list:

claim 10: see ¶32 of page 7 and ¶60 of page 16;

claim 12: see ¶39 and ¶40 of page 9;

claim 19: see ¶30 of pages 6-7;

claim 20: see ¶39 and ¶40 of page 9; and

claim 21: see ¶38 of pages 8-9.

Method claims 27, 28, 29, 23, 24 and 25 are similar to, respectively, apparatus claims 9, 10, 12, 19, 20 and 21. Therefore, method claims 27, 28, 29, 23, 24 and 25 are also allowable for at least the same reasons.

2. Claim Rejections – 35 USC § 103

The Examiner rejects claims 7-8 and 12 under 35 USC § 103(a) as being unpatentable over Butts in view of US Patent Application Publication 2002/0049578 (Ohkami). Please note that the Examiner erroneously refers to the Ohkami as "Tseng."

The Examiner bases her rejection mostly on Butts, referring to Ohkami just for purposes of illustrating data packet usage. Since claim 1 has been shown above to be patentable over Butts, claims 7-8 and 12, that depend on claim 1, must be patentable as well.

Furthermore, Ohkami relates to transfers of data, into and out of, a "design under verification" (DUV) by use of data packets. For example, ¶ 0019 of Ohkami states:

> The present invention uses a new packet-based protocol to perform data transfer operations between the host workstation and the hardware accelerator for loading data to and unloading data from the registers and memories in a target design under verification (also known as the target design or user's design).

Ohkami provides no teaching, suggestion or motivation that the data packet transfer technique could be applied to configuration of the DUV itself. Ohkami only addresses post-configuration processing. In fact, because Ohkami synthesizes at compile time the hardware used for packet processing, it actually teaches away from application of packet processing for purposes of configuration. For example, Ohkami states on page 2, ¶ 0021:

> In order to handle the request and response packets, a protocol interface logic is synthesized at compile time.

In addition, Ohkami only addresses the synthesis of special purpose hardware for the processing of data packets, and does not teach, suggest or motivate the use of a programmed processor. Figure 1 of Ohkami depicts a "hardware accelerator" 20, on which the packet processing hardware (called "protocol interface logic" 40) and the DUV 30 are synthesized. Within protocol interface logic 40 is depicted the next lower level of structure for the packet-handling hardware. The special purpose hardware of each functional block, within protocol interface logic 40, is depicted in the following figures of Ohkami: Figure 3 (for packet I/O block 41), Figure 4 (for memory address block 45), Figure 5 (for read data block 46), Figure 7 (for command decode block 43), Figure 9 (for state control block 42) and Figure 10 (for read/write control block 44).

3. Summary

Applicants respectfully submit that all 35 USC §102 and §103 rejections have been traversed. Therefore, applicants request a Notice of Allowance be granted.

The U.S. Patent and Trademark Office is hereby authorized to charge any fee deficiency, or credit any overpayment, to our Deposit Account No. 502584 referencing docket number MGC.00137.

Respectfully submitted,

/ Jonathan T. Kaplan /

Jonathan T. Kaplan (Reg. No. 38,935)

Date: July 09, 2007
Jonathan T. Kaplan
Attorney at Law
10800 SE 17th Circle, Suite E66
Vancouver, WA 98664-6297
Tel. 917-674-5017
JonathanKaplan@alum.MIT.edu